The Monkeytyping Solution to Challenge on Visual Understanding by Learning from Web Data

Ziheng Zhang Jia Zheng Shenghua Gao Yi Ma

School of Information Science and Technology
ShanghaiTech University

CVPR, 2017

Outline

Outline

Dataset Noise

Unlike manually annotated dataset such as ImageNet, the WebVision Dataset consists of 2.4 million of web images crawled from Internet, hence suffers from significant label noise as shown below.

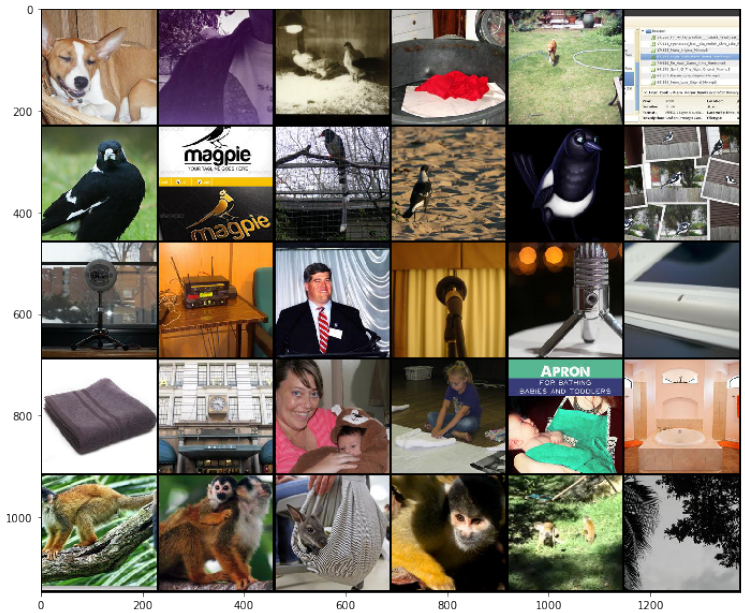


Figure: Some randomly chosen examples from five classes, examples in the same row belongs to the same class.

Dataset Noise

Besides, noise pattern in WebVision is relatively complicated. Some existing works either using simple noise model (eg. confusion matrix), or making strong assumptions on noise pattern (eg. random flip on labels with fixed noise level) do not work well on the WebVision dataset.

Figure: Noisy samples in category 'Tench'

Outline

Category Bias

The WebVision dataset also suffers from severe category bias. The number of images per category in the dataset is shown in the figure below, where we can see the number of samples varies from several hundreds to more than 10,000.

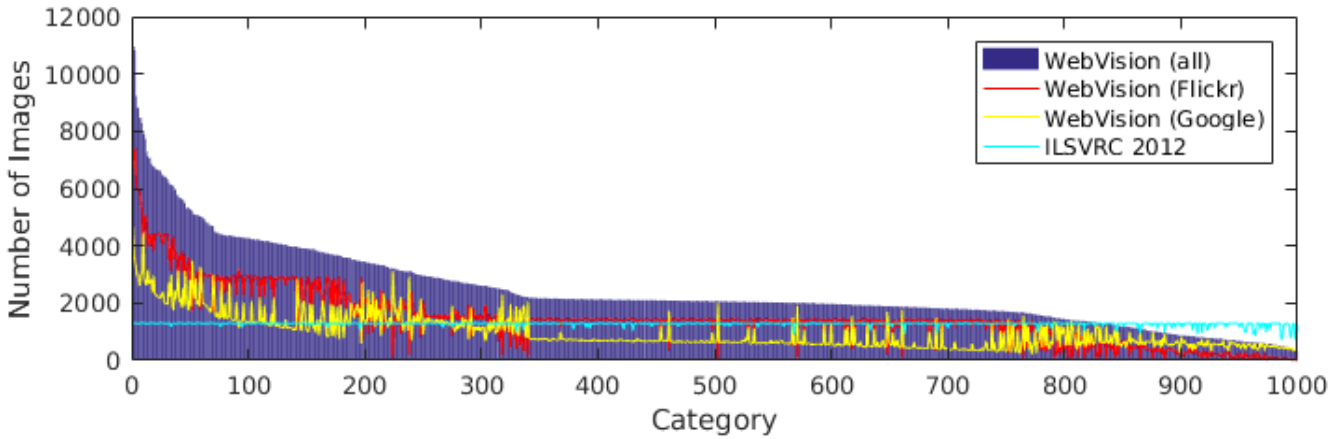


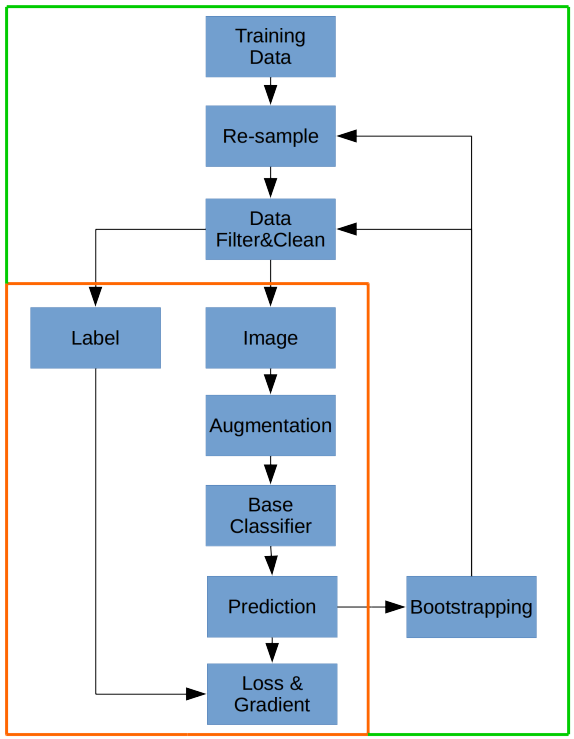
Figure: The number of images per category of the dataset.

Outline

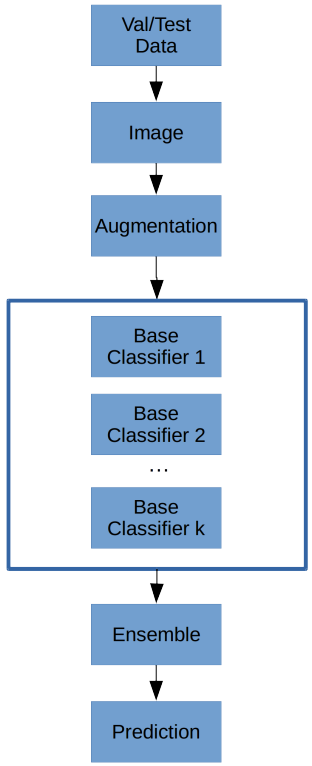
Solutions

In order to train a classifier using noisy web images, we design a series of training and testing procedure using techniques including **data re-sampling**, **bootstrapping** and **ensemble** (shown in the following slides). When training, the common training procedure (in orange box) is firstly applied several iterations, followed by the bootstrapping procedure (in greed box), which aims to clean the original noisy dataset. The two procedures run alternately. When testing, ensemble method are used to make the final prediction.

Solutions



(a) Training procedure



(b) Testing procedure

Outline

Base Classifier

We compared three state-of-art image classifiers, including ResNet, Inception-v4 and **Inception-ResNet-v2**. The last one achieves slightly better accuracy and converges faster than the other two classifiers.

Data Augmentation

We use the following data augmentation method after resizing them to a fixed size of 328×328 when training

- Random scale and aspect ratio augmentation[1]
- Color augmentation[2]
- Lighting[3]

After augmentation, we normalize each channel of image to zero mean and unit standard deviation.

---

[1] Christian Szegedy et al. "Going deeper with convolutions". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2015, pp. 1–9.

[2] Andrew G Howard. "Some improvements on deep convolutional neural network based image classification". In: *arXiv preprint arXiv:1312.5402* (2013).

[3] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. "Imagenet classification with deep convolutional neural networks". In: *Advances in neural information processing systems*. 2012, pp. 1097–1105.

Random scale and aspect ratio augmentation[4]

This method crops each image to a random size of 0.08 to 1.0 of the original size and then stretch it to a random aspect ratio of 3/4 to 4/3 of the original aspect ratio.

[4] Christian Szegedy et al. "Going deeper with convolutions". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2015, pp. 1–9.

Color augmentation[5]

This method changes the brightness, contrast and saturation of each image in random order to a random level of 0.6 to 1.4 of the original level.

[5] Andrew G Howard. "Some improvements on deep convolutional neural network based image classification". In: *arXiv preprint arXiv:1312.5402* (2013).

Lighting[6]

This method adds a noise vector drawn from a Gaussian. The direction of the Gaussian is same as that of the principal components of the dataset.

[6] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. "Imagenet classification with deep convolutional neural networks". In: *Advances in neural information processing systems*. 2012, pp. 1097–1105.

Outline

Re-sampling

In order to overcome category bias, we re-sample the dataset to guarantee that images of every class has the same probability to be sampled during training.

Data re-sampling improved the top-5 accuracy by about 1%.

Outline

Model Ensemble

We noticed that if we train the same base classifier several times, they have similar performance, but their performances on each class slightly vary.

Model Ensemble

Instead just average the output of several classifiers, we used an extra ensemble layer to learn the weights for each classifier and each class. Given classifier i and its predicted probability p_c^i for class c, our final class probability is for class c will be

$$p_c = \sum_i w_c^i p_c^i \tag{1}$$

where $w_c \in [0, 1]$, and $\sum_i w_c^i = 1$ are learnable parameters indicating to what extend should we believe the prediction of classifier i for class c.

We hope that the learnable ensemble can achieve better result than just do average, but find that they actually achieve a similar effect and improved the top-5 accuracy by about 3%.

Outline

Overcoming label noise

In order to deal with the label noise of the original training set, we use bootstrapping method[7] to remove or relabel the possible noisy samples.

The bootstrapping method improved the top-5 accuracy by about 2%.

[7] Scott Reed et al. "Training deep neural networks on noisy labels with bootstrapping". In: *arXiv preprint arXiv:1412.6596* (2014).

Relaxed bootstrapping

Input: training set
initialize an empty label mask table
foreach *image in training set* **do**
    predict label for the image
    if *predicted label equals to image label* **then**
        set mask label to image label
    else
        if *predicted probobality greater than a threshold* **then**
            set mask label to predicted label
        else
            set mask label to minus one
        end
    end
end
Output: label mask table

Algorithm 0: Bootstrapping Method

Outline

Training Procedure

Our training procedure consists of the following steps

1. Train five Inception-ResNet-v2 classifiers.
2. Train ensemble layer.
3. Use Bootstrapping method to clean the training set.
4. Fine-tune by cleaned training set.

Training Configuration

Table: Training Configuration

Learning Rate	0.15
Learning Rate Schedule	Decay by 0.1 every 30 epochs
Batch Size	300
Optimizer	SGD
Momentum	0.9

We train our models on 4 Tesla M40 GPUs in order to use larger batch size. We find that larger batch size will lead to better performance.

Outline

Testing Details

The 144 crops strategy[8] was used in testing.

[8] Christian Szegedy et al. "Going deeper with convolutions". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2015, pp. 1–9.

Outline

Final Submission

Table: Final Submission

Entry	Entry Description	Accuracy
1	Ensemble of classifiers with best top-1 accuracy	0.9223
2	Ensemble of classifiers with best top-5 accuracy	0.9225
3	Ensemble of classifiers with sub-best top-1 accuracy	0.9218
4	Ensemble of classifiers with sub-best top-5 accuracy	0.9218
5	The early result submitted during development phase	0.9216

Summary

Technically, our method contains nothing new, and we just try to find out the right way to learn from noisy web images using existing techniques. Here are what we have found when we deal with the WebVision Challenge

- Category bias in dataset can not be neglected, and data re-sampling is a good way to solve the problem.
- Bootstrapping method do overcome the dataset noise, hence significantly improve the classification performance.
- Generally, bigger models are more robust against dataset noise.
- It seems that bigger batch size leads to better result.

Thanks

Thanks for Attention!